

MAXREFDES117# Code Documentation

V01.00

Generated by Doxygen 1.8.2

Wed Jun 22 2016 15:24:07

Contents

1	Main Page	1
1.1	Introduction	1
2	File Index	3
2.1	File List	3
3	File Documentation	5
3.1	RD117_MBED/algorithm/algorithm.cpp File Reference	5
3.1.1	Detailed Description	5
3.1.2	Function Documentation	6
3.1.2.1	maxim_find_peaks	6
3.1.2.2	maxim_heart_rate_and_oxygen_saturation	6
3.1.2.3	maxim_peaks_above_min_height	6
3.1.2.4	maxim_remove_close_peaks	7
3.1.2.5	maxim_sort_ascend	7
3.1.2.6	maxim_sort_indices_descend	7
3.2	RD117_MBED/algorithm/algorithm.h File Reference	8
3.2.1	Detailed Description	9
3.2.2	Macro Definition Documentation	9
3.2.2.1	BUFFER_SIZE	9
3.2.2.2	false	9
3.2.2.3	FS	9
3.2.2.4	HAMMING_SIZE	9
3.2.2.5	HR_FIFO_SIZE	10
3.2.2.6	MA4_SIZE	10
3.2.2.7	min	10
3.2.2.8	true	10
3.2.3	Function Documentation	10
3.2.3.1	maxim_find_peaks	10

3.2.3.2	maxim_heart_rate_and_oxygen_saturation	10
3.2.3.3	maxim_peaks_above_min_height	11
3.2.3.4	maxim_remove_close_peaks	11
3.2.3.5	maxim_sort_ascend	11
3.2.3.6	maxim_sort_indices_descend	12
3.2.4	Variable Documentation	12
3.2.4.1	auw_hamm	12
3.2.4.2	uch_spo2_table	12
3.3	RD117_MBED/main.cpp File Reference	12
3.3.1	Detailed Description	13
3.3.2	Macro Definition Documentation	13
3.3.2.1	MAX_BRIGHTNESS	13
3.3.3	Function Documentation	13
3.3.3.1	main	13
3.3.3.2	pc	14
3.3.4	Variable Documentation	14
3.3.4.1	aun_ir_buffer	14
3.3.4.2	aun_red_buffer	14
3.3.4.3	ch_hr_valid	14
3.3.4.4	ch_spo2_valid	14
3.3.4.5	n_heart_rate	14
3.3.4.6	n_ir_buffer_length	14
3.3.4.7	n_sp02	14
3.3.4.8	uch_dummy	14
3.4	RD117_MBED/MAX30102/MAX30102.cpp File Reference	14
3.4.1	Detailed Description	15
3.4.2	Function Documentation	15
3.4.2.1	i2c	15
3.4.2.2	maxim_max30102_init	15
3.4.2.3	maxim_max30102_read_fifo	15
3.4.2.4	maxim_max30102_read_reg	16
3.4.2.5	maxim_max30102_reset	16
3.4.2.6	maxim_max30102_write_reg	17
3.5	RD117_MBED/MAX30102/MAX30102.h File Reference	17
3.5.1	Detailed Description	18
3.5.2	Macro Definition Documentation	18
3.5.2.1	I2C_READ_ADDR	18

3.5.2.2	I2C_WRITE_ADDR	18
3.5.2.3	REG_FIFO_CONFIG	18
3.5.2.4	REG_FIFO_DATA	18
3.5.2.5	REG_FIFO_RD_PTR	18
3.5.2.6	REG_FIFO_WR_PTR	19
3.5.2.7	REG_INTR_ENABLE_1	19
3.5.2.8	REG_INTR_ENABLE_2	19
3.5.2.9	REG_INTR_STATUS_1	19
3.5.2.10	REG_INTR_STATUS_2	19
3.5.2.11	REG_LED1_PA	19
3.5.2.12	REG_LED2_PA	19
3.5.2.13	REG_MODE_CONFIG	19
3.5.2.14	REG_MULTI_LED_CTRL1	19
3.5.2.15	REG_MULTI_LED_CTRL2	19
3.5.2.16	REG_OVF_COUNTER	19
3.5.2.17	REG_PART_ID	20
3.5.2.18	REG_PILOT_PA	20
3.5.2.19	REG_PROX_INT_THRESH	20
3.5.2.20	REG_REV_ID	20
3.5.2.21	REG_SPO2_CONFIG	20
3.5.2.22	REG_TEMP_CONFIG	20
3.5.2.23	REG_TEMP_FRAC	20
3.5.2.24	REG_TEMP_INTR	20
3.5.3	Function Documentation	20
3.5.3.1	maxim_max30102_init	20
3.5.3.2	maxim_max30102_read_fifo	21
3.5.3.3	maxim_max30102_read_reg	21
3.5.3.4	maxim_max30102_reset	21
3.5.3.5	maxim_max30102_write_reg	22

Chapter 1

Main Page

1.1 Introduction

This is the code documentation for the MAXREFDES117# subsystem reference design.

The Files page contains the File List page and the Globals page.

The Globals page contains the Functions, Variables, and Macros sub-pages.

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

RD117_MBED/ main.cpp	12
RD117_MBED/algorithm/ algorithm.cpp	5
RD117_MBED/algorithm/ algorithm.h	8
RD117_MBED/MAX30102/ MAX30102.cpp	14
RD117_MBED/MAX30102/ MAX30102.h	17

Chapter 3

File Documentation

3.1 RD117_MBED/algorithm/algorithm.cpp File Reference

```
#include "algorithm.h"
#include "mbed.h"
```

Functions

- void [maxim_heart_rate_and_oxygen_saturation](#) (uint32_t *pun_ir_buffer, int32_t [n_ir_buffer_length](#), uint32_t *pun_red_buffer, int32_t *pn_spo2, int8_t *pch_spo2_valid, int32_t *pn_heart_rate, int8_t *pch_hr_valid)
Calculate the heart rate and SpO2 level.
- void [maxim_find_peaks](#) (int32_t *pn_locs, int32_t *pn_npk, int32_t *pn_x, int32_t n_size, int32_t n_min_height, int32_t n_min_distance, int32_t n_max_num)
Find peaks.
- void [maxim_peaks_above_min_height](#) (int32_t *pn_locs, int32_t *pn_npk, int32_t *pn_x, int32_t n_size, int32_t n_min_height)
Find peaks above n_min_height.
- void [maxim_remove_close_peaks](#) (int32_t *pn_locs, int32_t *pn_npk, int32_t *pn_x, int32_t n_min_distance)
Remove peaks.
- void [maxim_sort_ascend](#) (int32_t *pn_x, int32_t n_size)
Sort array.
- void [maxim_sort_indices_descend](#) (int32_t *pn_x, int32_t *pn_idx, int32_t n_size)
Sort indices.

3.1.1 Detailed Description

Project: MAXREFDES117# Filename: [algorithm.cpp](#) Description: This module calculates the heart rate/SpO2 level

This code follows the following naming conventions:

char ch_pmod_value char (array) s_pmod_s_string[16] float f_pmod_value int32_t n_pmod_value int32_t (array) an_pmod_value[16] int16_t w_pmod_value int16_t (array) aw_pmod_value[16] uint16_t uw_pmod_value uint16_t (array) auw_pmod_value[16] uint8_t uch_pmod_value uint8_t (array) auch_pmod_buffer[16] uint32_t un_pmod_value int32_t * pn_pmod_value

Definition in file [algorithm.cpp](#).

3.1.2 Function Documentation

3.1.2.1 void maxim_find_peaks (int32_t * pn_locs, int32_t * pn_npk, int32_t * pn_x, int32_t n_size, int32_t n_min_height, int32_t n_min_distance, int32_t n_max_num)

Find peaks.

Details

Find at most MAX_NUM peaks above MIN_HEIGHT separated by at least MIN_DISTANCE

Return values

None

Definition at line 254 of file algorithm.cpp.

3.1.2.2 void maxim_heart_rate_and_oxygen_saturation (uint32_t * pun_ir_buffer, int32_t n_ir_buffer_length, uint32_t * pun_red_buffer, int32_t * pn_spo2, int8_t * pch_spo2_valid, int32_t * pn_heart_rate, int8_t * pch_hr_valid)

Calculate the heart rate and SpO2 level.

Details

By detecting peaks of PPG cycle and corresponding AC/DC of red/infra-red signal, the ratio for the SPO2 is computed. Since this algorithm is aiming for Arm M0/M3. formula for SPO2 did not achieve the accuracy due to register overflow. Thus, accurate SPO2 is precalculated and save longo uch_spo2_table[] per each ratio.

Parameters

in	*pun_ir_buffer	- IR sensor data buffer
in	n_ir_buffer_length	- IR sensor data buffer length
in	*pun_red_buffer	- Red sensor data buffer
out	*pn_spo2	- Calculated SpO2 value
out	*pch_spo2_valid	- 1 if the calculated SpO2 value is valid
out	*pn_heart_rate	- Calculated heart rate value
out	*pch_hr_valid	- 1 if the calculated heart rate value is valid

Return values

None

Definition at line 62 of file algorithm.cpp.

3.1.2.3 void maxim_peaks_above_min_height (int32_t * pn_locs, int32_t * pn_npk, int32_t * pn_x, int32_t n_size, int32_t n_min_height)

Find peaks above n_min_height.

Details

Find all peaks above MIN_HEIGHT

Return values

<i>None</i>	
-------------	--

Definition at line 268 of file algorithm.cpp.

3.1.2.4 void maxim_remove_close_peaks (int32_t * *pn_locs*, int32_t * *pn_npks*, int32_t * *pn_x*, int32_t *n_min_distance*)

Remove peaks.

Details

Remove peaks separated by less than MIN_DISTANCE

Return values

<i>None</i>	
-------------	--

Definition at line 299 of file algorithm.cpp.

3.1.2.5 void maxim_sort_ascend (int32_t * *pn_x*, int32_t *n_size*)

Sort array.

Details

Sort array in ascending order (insertion sort algorithm)

Return values

<i>None</i>	
-------------	--

Definition at line 328 of file algorithm.cpp.

3.1.2.6 void maxim_sort_indices_descend (int32_t * *pn_x*, int32_t * *pn_indx*, int32_t *n_size*)

Sort indices.

Details

Sort indices according to descending order (insertion sort algorithm)

Return values

<i>None</i>	
-------------	--

Definition at line 346 of file algorithm.cpp.

3.2 RD117_MBED/algorithm/algorithm.h File Reference

```
#include "mbed.h"
```

Macros

- `#define true 1`
- `#define false 0`
- `#define FS 100`
- `#define BUFFER_SIZE (FS* 5)`
- `#define HR_FIFO_SIZE 7`
- `#define MA4_SIZE 4`
- `#define HAMMING_SIZE 5`
- `#define min(x, y) ((x) < (y) ? (x) : (y))`

Functions

- void `maxim_heart_rate_and_oxygen_saturation` (uint32_t *pun_ir_buffer, int32_t n_ir_buffer_length, uint32_t *pun_red_buffer, int32_t *pn_spo2, int8_t *pch_spo2_valid, int32_t *pn_heart_rate, int8_t *pch_hr_valid)
Calculate the heart rate and SpO2 level.
- void `maxim_find_peaks` (int32_t *pn_locs, int32_t *pn_npk, int32_t *pn_x, int32_t n_size, int32_t n_min_height, int32_t n_min_distance, int32_t n_max_num)
Find peaks.
- void `maxim_peaks_above_min_height` (int32_t *pn_locs, int32_t *pn_npk, int32_t *pn_x, int32_t n_size, int32_t n_min_height)
Find peaks above n_min_height.
- void `maxim_remove_close_peaks` (int32_t *pn_locs, int32_t *pn_npk, int32_t *pn_x, int32_t n_min_distance)
Remove peaks.
- void `maxim_sort_ascend` (int32_t *pn_x, int32_t n_size)
Sort array.
- void `maxim_sort_indices_descend` (int32_t *pn_x, int32_t *pn_idx, int32_t n_size)
Sort indices.

Variables

- const uint16_t `auw_hamm` [31] = { 41, 276, 512, 276, 41 }
- const uint8_t `uch_spo2_table` [184]

3.2.1 Detailed Description

Project: MAXREFDES117# Filename: [algorithm.h](#) Description: This module is the heart rate/SpO2 calculation algorithm header file

Revision History:

1-18-2016 Rev 01.00 SK Initial release.

This code follows the following naming conventions:

char ch_pmod_value

char (array) s_pmod_s_string[16]

float f_pmod_value

int32_t n_pmod_value

int32_t (array) an_pmod_value[16]

int16_t w_pmod_value

int16_t (array) aw_pmod_value[16]

uint16_t uw_pmod_value

uint16_t (array) auw_pmod_value[16]

uint8_t uch_pmod_value

uint8_t (array) auch_pmod_buffer[16]

uint32_t un_pmod_value

int32_t * pn_pmod_value

Definition in file [algorithm.h](#).

3.2.2 Macro Definition Documentation

3.2.2.1 #define BUFFER_SIZE (FS* 5)

Definition at line 70 of file [algorithm.h](#).

3.2.2.2 #define false 0

Definition at line 68 of file [algorithm.h](#).

3.2.2.3 #define FS 100

Definition at line 69 of file [algorithm.h](#).

3.2.2.4 #define HAMMING_SIZE 5

Definition at line 73 of file [algorithm.h](#).

3.2.2.5 #define HR_FIFO_SIZE 7

Definition at line 71 of file algorithm.h.

3.2.2.6 #define MA4_SIZE 4

Definition at line 72 of file algorithm.h.

3.2.2.7 #define min(x, y) ((x) < (y) ? (x) : (y))

Definition at line 74 of file algorithm.h.

3.2.2.8 #define true 1

Definition at line 67 of file algorithm.h.

3.2.3 Function Documentation**3.2.3.1 void maxim_find_peaks (int32_t * pn_locs, int32_t * pn_npk, int32_t * pn_x, int32_t n_size, int32_t n_min_height, int32_t n_min_distance, int32_t n_max_num)**

Find peaks.

Details

Find at most MAX_NUM peaks above MIN_HEIGHT separated by at least MIN_DISTANCE

Return values

<i>None</i>	
-------------	--

Definition at line 254 of file algorithm.cpp.

3.2.3.2 void maxim_heart_rate_and_oxygen_saturation (uint32_t * pun_ir_buffer, int32_t n_ir_buffer_length, uint32_t * pun_red_buffer, int32_t * pn_spo2, int8_t * pch_spo2_valid, int32_t * pn_heart_rate, int8_t * pch_hr_valid)

Calculate the heart rate and SpO2 level.

Details

By detecting peaks of PPG cycle and corresponding AC/DC of red/infra-red signal, the ratio for the SPO2 is computed. Since this algorithm is aiming for Arm M0/M3. formula for SPO2 did not achieve the accuracy due to register overflow. Thus, accurate SPO2 is precalculated and save longo uch_spo2_table[] per each ratio.

Parameters

in	<i>*pun_ir_buffer</i>	- IR sensor data buffer
in	<i>n_ir_buffer_length</i>	- IR sensor data buffer length

in	<i>*pun_red_buffer</i>	- Red sensor data buffer
out	<i>*pn_spo2</i>	- Calculated SpO2 value
out	<i>*pch_spo2_valid</i>	- 1 if the calculated SpO2 value is valid
out	<i>*pn_heart_rate</i>	- Calculated heart rate value
out	<i>*pch_hr_valid</i>	- 1 if the calculated heart rate value is valid

Return values

<i>None</i>

Definition at line 62 of file algorithm.cpp.

3.2.3.3 void maxim_peaks_above_min_height (int32_t * *pn_locs*, int32_t * *pn_npks*, int32_t * *pn_x*, int32_t *n_size*, int32_t *n_min_height*)

Find peaks above *n_min_height*.

Details

Find all peaks above MIN_HEIGHT

Return values

<i>None</i>

Definition at line 268 of file algorithm.cpp.

3.2.3.4 void maxim_remove_close_peaks (int32_t * *pn_locs*, int32_t * *pn_npks*, int32_t * *pn_x*, int32_t *n_min_distance*)

Remove peaks.

Details

Remove peaks separated by less than MIN_DISTANCE

Return values

<i>None</i>

Definition at line 299 of file algorithm.cpp.

3.2.3.5 void maxim_sort_ascend (int32_t * *pn_x*, int32_t *n_size*)

Sort array.

Details

Sort array in ascending order (insertion sort algorithm)

Return values

<i>None</i>

Definition at line 328 of file algorithm.cpp.

3.2.3.6 void maxim_sort_indices_descend (int32_t * pn_x, int32_t * pn_indx, int32_t n_size)

Sort indices.

Details

Sort indices according to descending order (insertion sort algorithm)

Return values

<i>None</i>

Definition at line 346 of file algorithm.cpp.

3.2.4 Variable Documentation**3.2.4.1 const uint16_t auw_hamm[31] = { 41, 276, 512, 276, 41 }**

Definition at line 76 of file algorithm.h.

3.2.4.2 const uint8_t uch_spo2_table[184]**Initial value:**

```
= { 95, 95, 95, 96, 96, 96, 97, 97, 97, 97, 97, 98, 98, 98, 98, 98, 99, 99, 99,
    99,
    99, 99, 99, 99, 100, 100, 100, 100, 100, 100, 100, 100,
    100, 100, 100, 100, 100, 100, 100, 100, 100,
    100, 100, 100, 100, 99, 99, 99, 99, 99, 99, 99, 99, 99,
    98, 98, 98, 98, 98, 98, 97, 97,
    97, 97, 96, 96, 96, 96, 95, 95, 95, 94, 94, 94, 93,
    93, 93, 92, 92, 92, 91, 91,
    90, 90, 89, 89, 89, 88, 88, 87, 87, 86, 86, 85, 85,
    84, 84, 83, 82, 82, 81, 81,
    80, 80, 79, 78, 78, 77, 76, 76, 75, 74, 74, 73, 72,
    72, 71, 70, 69, 69, 68, 67,
    66, 66, 65, 64, 63, 62, 62, 61, 60, 59, 58, 57, 56,
    56, 55, 54, 53, 52, 51, 50,
    49, 48, 47, 46, 45, 44, 43, 42, 41, 40, 39, 38, 37,
    36, 35, 34, 33, 31, 30, 29,
    28, 27, 26, 25, 23, 22, 21, 20, 19, 17, 16, 15, 14,
    12, 11, 10, 9, 7, 6, 5,
    3, 2, 1 }
```

Definition at line 78 of file algorithm.h.

3.3 RD117_MBED/main.cpp File Reference

```
#include "mbed.h"
#include "algorithm.h"
#include "MAX30102.h"
```

Macros

- #define [MAX_BRIGHTNESS](#) 255

Functions

- Serial [pc](#) (USBTX, USBRX)
- int [main](#) ()

Variables

- uint32_t [aun_ir_buffer](#) [500]
- int32_t [n_ir_buffer_length](#)
- uint32_t [aun_red_buffer](#) [500]
- int32_t [n_sp02](#)
- int8_t [ch_spo2_valid](#)
- int32_t [n_heart_rate](#)
- int8_t [ch_hr_valid](#)
- uint8_t [uch_dummy](#)

3.3.1 Detailed Description

Project: MAXREFDES117# Filename: [main.cpp](#) Description: This module contains the Main application for the MAXREFDES117 example program.

This code follows the following naming conventions:

char ch_pmod_value char (array) s_pmod_s_string[16] float f_pmod_value int32_t n_pmod_value int32_t (array) an_pmod_value[16] int16_t w_pmod_value int16_t (array) aw_pmod_value[16] uint16_t uw_pmod_value uint16_t (array) auw_pmod_value[16] uint8_t uch_pmod_value uint8_t (array) auch_pmod_buffer[16] uint32_t un_pmod_value int32_t * pn_pmod_value

Definition in file [main.cpp](#).

3.3.2 Macro Definition Documentation

3.3.2.1 #define MAX_BRIGHTNESS 255

Definition at line 78 of file [main.cpp](#).

3.3.3 Function Documentation

3.3.3.1 int main ()

Definition at line 103 of file [main.cpp](#).

3.3.3.2 Serial pc (USBTX , USBRX)

3.3.4 Variable Documentation

3.3.4.1 uint32_t aun_ir_buffer[500]

Definition at line 80 of file main.cpp.

3.3.4.2 uint32_t aun_red_buffer[500]

Definition at line 82 of file main.cpp.

3.3.4.3 int8_t ch_hr_valid

Definition at line 86 of file main.cpp.

3.3.4.4 int8_t ch_spo2_valid

Definition at line 84 of file main.cpp.

3.3.4.5 int32_t n_heart_rate

Definition at line 85 of file main.cpp.

3.3.4.6 int32_t n_ir_buffer_length

Definition at line 81 of file main.cpp.

3.3.4.7 int32_t n_sp02

Definition at line 83 of file main.cpp.

3.3.4.8 uint8_t uch_dummy

Definition at line 87 of file main.cpp.

3.4 RD117_MBED/MAX30102/MAX30102.cpp File Reference

```
#include "mbed.h"  
#include "MAX30102.h"
```

Functions

- I2C [i2c](#) (I2C_SDA, I2C_SCL)
- bool [maxim_max30102_write_reg](#) (uint8_t uch_addr, uint8_t uch_data)

Write a value to a MAX30102 register.

- bool [maxim_max30102_read_reg](#) (uint8_t uch_addr, uint8_t *puch_data)

Read a MAX30102 register.

- bool [maxim_max30102_init](#) ()

Initialize the MAX30102.

- bool [maxim_max30102_read_fifo](#) (uint32_t *pun_red_led, uint32_t *pun_ir_led)

Read a set of samples from the MAX30102 FIFO register.

- bool [maxim_max30102_reset](#) ()

Reset the MAX30102.

3.4.1 Detailed Description

Project: MAXREFDES117# Filename: [max30102.cpp](#) Description: This module is an embedded controller driver for the MAX30102

This code follows the following naming conventions:

char ch_pmod_value char (array) s_pmod_s_string[16] float f_pmod_value int32_t n_pmod_value int32_t (array) an_pmod_value[16] int16_t w_pmod_value int16_t (array) aw_pmod_value[16] uint16_t uw_pmod_value uint16_t (array) auw_pmod_value[16] uint8_t uch_pmod_value uint8_t (array) auch_pmod_buffer[16] uint32_t un_pmod_value int32_t * pn_pmod_value

Definition in file [MAX30102.cpp](#).

3.4.2 Function Documentation

3.4.2.1 I2C i2c (I2C_SDA , I2C_SCL)

3.4.2.2 bool maxim_max30102_init ()

Initialize the MAX30102.

Details

This function initializes the MAX30102

Parameters

<i>None</i>	
-------------	--

Return values

<i>true</i>	on success
-------------	------------

Definition at line 115 of file MAX30102.cpp.

3.4.2.3 bool maxim_max30102_read_fifo (uint32_t * pun_red_led, uint32_t * pun_ir_led)

Read a set of samples from the MAX30102 FIFO register.

Details

This function reads a set of samples from the MAX30102 FIFO register

Parameters

out	<i>*pun_red_led</i>	- pointer that stores the red LED reading data
out	<i>*pun_ir_led</i>	- pointer that stores the IR LED reading data

Return values

<i>true</i>	on success
-------------	------------

Definition at line 152 of file MAX30102.cpp.

3.4.2.4 bool maxim_max30102_read_reg (uint8_t uch_addr, uint8_t * puch_data)

Read a MAX30102 register.

Details

This function reads a MAX30102 register

Parameters

in	<i>uch_addr</i>	- register address
out	<i>puch_data</i>	- pointer that stores the register data

Return values

<i>true</i>	on success
-------------	------------

Definition at line 90 of file MAX30102.cpp.

3.4.2.5 bool maxim_max30102_reset (void)

Reset the MAX30102.

Details

This function resets the MAX30102

Parameters

<i>None</i>	
-------------	--

Return values

<i>true</i>	on success
-------------	------------

Definition at line 205 of file MAX30102.cpp.

3.4.2.6 bool maxim_max30102_write_reg (uint8_t uch_addr, uint8_t uch_data)

Write a value to a MAX30102 register.

Details

This function writes a value to a MAX30102 register

Parameters

in	<i>uch_addr</i>	- register address
in	<i>uch_data</i>	- register data

Return values

<i>true</i>	on success
-------------	------------

Definition at line 68 of file MAX30102.cpp.

3.5 RD117_MBED/MAX30102/MAX30102.h File Reference

```
#include "mbed.h"
```

Macros

- #define I2C_WRITE_ADDR 0xAE
- #define I2C_READ_ADDR 0xAF
- #define REG_INTR_STATUS_1 0x00
- #define REG_INTR_STATUS_2 0x01
- #define REG_INTR_ENABLE_1 0x02
- #define REG_INTR_ENABLE_2 0x03
- #define REG_FIFO_WR_PTR 0x04
- #define REG_OVF_COUNTER 0x05
- #define REG_FIFO_RD_PTR 0x06
- #define REG_FIFO_DATA 0x07
- #define REG_FIFO_CONFIG 0x08
- #define REG_MODE_CONFIG 0x09
- #define REG_SPO2_CONFIG 0x0A
- #define REG_LED1_PA 0x0C
- #define REG_LED2_PA 0x0D
- #define REG_PILOT_PA 0x10
- #define REG_MULTI_LED_CTRL1 0x11
- #define REG_MULTI_LED_CTRL2 0x12
- #define REG_TEMP_INTR 0x1F
- #define REG_TEMP_FRAC 0x20
- #define REG_TEMP_CONFIG 0x21
- #define REG_PROX_INT_THRESH 0x30
- #define REG_REV_ID 0xFE
- #define REG_PART_ID 0xFF

Functions

- bool [maxim_max30102_init](#) ()
Initialize the MAX30102.
- bool [maxim_max30102_read_fifo](#) (uint32_t *pun_red_led, uint32_t *pun_ir_led)
Read a set of samples from the MAX30102 FIFO register.
- bool [maxim_max30102_write_reg](#) (uint8_t uch_addr, uint8_t uch_data)
Write a value to a MAX30102 register.
- bool [maxim_max30102_read_reg](#) (uint8_t uch_addr, uint8_t *puch_data)
Read a MAX30102 register.
- bool [maxim_max30102_reset](#) (void)
Reset the MAX30102.

3.5.1 Detailed Description

Project: MAXREFDES117# Filename: [max30102.h](#) Description: This module is an embedded controller driver header file for MAX30102

This code follows the following naming conventions:

char ch_pmod_value char (array) s_pmod_s_string[16] float f_pmod_value int32_t n_pmod_value int32_t (array) an_pmod_value[16] int16_t w_pmod_value int16_t (array) aw_pmod_value[16] uint16_t uw_pmod_value uint16_t (array) auw_pmod_value[16] uint8_t uch_pmod_value uint8_t (array) auch_pmod_buffer[16] uint32_t un_pmod_value int32_t * pn_pmod_value

Definition in file [MAX30102.h](#).

3.5.2 Macro Definition Documentation

3.5.2.1 #define I2C_READ_ADDR 0xAF

Definition at line 65 of file MAX30102.h.

3.5.2.2 #define I2C_WRITE_ADDR 0xAE

Definition at line 64 of file MAX30102.h.

3.5.2.3 #define REG_FIFO_CONFIG 0x08

Definition at line 76 of file MAX30102.h.

3.5.2.4 #define REG_FIFO_DATA 0x07

Definition at line 75 of file MAX30102.h.

3.5.2.5 #define REG_FIFO_RD_PTR 0x06

Definition at line 74 of file MAX30102.h.

3.5.2.6 #define REG_FIFO_WR_PTR 0x04

Definition at line 72 of file MAX30102.h.

3.5.2.7 #define REG_INTR_ENABLE_1 0x02

Definition at line 70 of file MAX30102.h.

3.5.2.8 #define REG_INTR_ENABLE_2 0x03

Definition at line 71 of file MAX30102.h.

3.5.2.9 #define REG_INTR_STATUS_1 0x00

Definition at line 68 of file MAX30102.h.

3.5.2.10 #define REG_INTR_STATUS_2 0x01

Definition at line 69 of file MAX30102.h.

3.5.2.11 #define REG_LED1_PA 0x0C

Definition at line 79 of file MAX30102.h.

3.5.2.12 #define REG_LED2_PA 0x0D

Definition at line 80 of file MAX30102.h.

3.5.2.13 #define REG_MODE_CONFIG 0x09

Definition at line 77 of file MAX30102.h.

3.5.2.14 #define REG_MULTI_LED_CTRL1 0x11

Definition at line 82 of file MAX30102.h.

3.5.2.15 #define REG_MULTI_LED_CTRL2 0x12

Definition at line 83 of file MAX30102.h.

3.5.2.16 #define REG_OVF_COUNTER 0x05

Definition at line 73 of file MAX30102.h.

3.5.2.17 #define REG_PART_ID 0xFF

Definition at line 89 of file MAX30102.h.

3.5.2.18 #define REG_PILOT_PA 0x10

Definition at line 81 of file MAX30102.h.

3.5.2.19 #define REG_PROX_INT_THRESH 0x30

Definition at line 87 of file MAX30102.h.

3.5.2.20 #define REG_REV_ID 0xFE

Definition at line 88 of file MAX30102.h.

3.5.2.21 #define REG_SPO2_CONFIG 0x0A

Definition at line 78 of file MAX30102.h.

3.5.2.22 #define REG_TEMP_CONFIG 0x21

Definition at line 86 of file MAX30102.h.

3.5.2.23 #define REG_TEMP_FRAC 0x20

Definition at line 85 of file MAX30102.h.

3.5.2.24 #define REG_TEMP_INTR 0x1F

Definition at line 84 of file MAX30102.h.

3.5.3 Function Documentation**3.5.3.1 bool maxim_max30102_init ()**

Initialize the MAX30102.

Details

This function initializes the MAX30102

Parameters

<i>None</i>	
-------------	--

Return values

<i>true</i>	on success
-------------	------------

Definition at line 115 of file MAX30102.cpp.

3.5.3.2 bool maxim_max30102_read_fifo (uint32_t * *pun_red_led*, uint32_t * *pun_ir_led*)

Read a set of samples from the MAX30102 FIFO register.

Details

This function reads a set of samples from the MAX30102 FIFO register

Parameters

out	<i>*pun_red_led</i>	- pointer that stores the red LED reading data
out	<i>*pun_ir_led</i>	- pointer that stores the IR LED reading data

Return values

<i>true</i>	on success
-------------	------------

Definition at line 152 of file MAX30102.cpp.

3.5.3.3 bool maxim_max30102_read_reg (uint8_t *uch_addr*, uint8_t * *puch_data*)

Read a MAX30102 register.

Details

This function reads a MAX30102 register

Parameters

in	<i>uch_addr</i>	- register address
out	<i>puch_data</i>	- pointer that stores the register data

Return values

<i>true</i>	on success
-------------	------------

Definition at line 90 of file MAX30102.cpp.

3.5.3.4 bool maxim_max30102_reset (void)

Reset the MAX30102.

Details

This function resets the MAX30102

Parameters

<i>None</i>	
-------------	--

Return values

<i>true</i>	on success
-------------	------------

Definition at line 205 of file MAX30102.cpp.

3.5.3.5 bool maxim_max30102_write_reg (uint8_t *uch_addr*, uint8_t *uch_data*)

Write a value to a MAX30102 register.

Details

This function writes a value to a MAX30102 register

Parameters

in	<i>uch_addr</i>	- register address
in	<i>uch_data</i>	- register data

Return values

<i>true</i>	on success
-------------	------------

Definition at line 68 of file MAX30102.cpp.

Index

algorithm.cpp
 maxim_find_peaks, 6
 maxim_heart_rate_and_oxygen_saturation, 6
 maxim_peaks_above_min_height, 6
 maxim_remove_close_peaks, 7
 maxim_sort_ascend, 7
 maxim_sort_indices_descend, 7

algorithm.h
 auw_hamm, 12
 BUFFER_SIZE, 9
 FS, 9
 false, 9
 HAMMING_SIZE, 9
 HR_FIFO_SIZE, 9
 MA4_SIZE, 10
 maxim_find_peaks, 10
 maxim_heart_rate_and_oxygen_saturation, 10
 maxim_peaks_above_min_height, 11
 maxim_remove_close_peaks, 11
 maxim_sort_ascend, 11
 maxim_sort_indices_descend, 12
 min, 10
 true, 10
 uch_spo2_table, 12

aun_ir_buffer
 main.cpp, 14

aun_red_buffer
 main.cpp, 14

auw_hamm
 algorithm.h, 12

BUFFER_SIZE
 algorithm.h, 9

ch_hr_valid
 main.cpp, 14

ch_spo2_valid
 main.cpp, 14

FS
 algorithm.h, 9

false
 algorithm.h, 9

HAMMING_SIZE
 algorithm.h, 9

HR_FIFO_SIZE
 algorithm.h, 9

I2C_READ_ADDR
 MAX30102.h, 18

I2C_WRITE_ADDR
 MAX30102.h, 18

i2c
 MAX30102.cpp, 15

MA4_SIZE
 algorithm.h, 10

MAX30102.cpp
 i2c, 15
 maxim_max30102_init, 15
 maxim_max30102_read_fifo, 15
 maxim_max30102_read_reg, 16
 maxim_max30102_reset, 16
 maxim_max30102_write_reg, 16

MAX30102.h
 I2C_READ_ADDR, 18
 I2C_WRITE_ADDR, 18
 maxim_max30102_init, 20
 maxim_max30102_read_fifo, 21
 maxim_max30102_read_reg, 21
 maxim_max30102_reset, 21
 maxim_max30102_write_reg, 22
 REG_FIFO_CONFIG, 18
 REG_FIFO_DATA, 18
 REG_FIFO_RD_PTR, 18
 REG_FIFO_WR_PTR, 18
 REG_INTR_ENABLE_1, 19
 REG_INTR_ENABLE_2, 19
 REG_INTR_STATUS_1, 19
 REG_INTR_STATUS_2, 19
 REG_LED1_PA, 19
 REG_LED2_PA, 19
 REG_MODE_CONFIG, 19
 REG_OVF_COUNTER, 19
 REG_PART_ID, 19
 REG_PILOT_PA, 20
 REG_REV_ID, 20
 REG_SPO2_CONFIG, 20
 REG_TEMP_CONFIG, 20
 REG_TEMP_FRAC, 20
 REG_TEMP_INTR, 20

MAX_BRIGHTNESS
 main.cpp, 13
 main
 main.cpp, 13
 main.cpp
 aun_ir_buffer, 14
 aun_red_buffer, 14
 ch_hr_valid, 14
 ch_spo2_valid, 14
 MAX_BRIGHTNESS, 13
 main, 13
 n_heart_rate, 14
 n_ir_buffer_length, 14
 n_spo2, 14
 pc, 13
 uch_dummy, 14
 maxim_find_peaks
 algorithm.cpp, 6
 algorithm.h, 10
 maxim_heart_rate_and_oxygen_saturation
 algorithm.cpp, 6
 algorithm.h, 10
 maxim_max30102_init
 MAX30102.cpp, 15
 MAX30102.h, 20
 maxim_max30102_read_fifo
 MAX30102.cpp, 15
 MAX30102.h, 21
 maxim_max30102_read_reg
 MAX30102.cpp, 16
 MAX30102.h, 21
 maxim_max30102_reset
 MAX30102.cpp, 16
 MAX30102.h, 21
 maxim_max30102_write_reg
 MAX30102.cpp, 16
 MAX30102.h, 22
 maxim_peaks_above_min_height
 algorithm.cpp, 6
 algorithm.h, 11
 maxim_remove_close_peaks
 algorithm.cpp, 7
 algorithm.h, 11
 maxim_sort_ascend
 algorithm.cpp, 7
 algorithm.h, 11
 maxim_sort_indices_descend
 algorithm.cpp, 7
 algorithm.h, 12
 min
 algorithm.h, 10

 n_heart_rate
 main.cpp, 14

 n_ir_buffer_length
 main.cpp, 14
 n_spo2
 main.cpp, 14

 pc
 main.cpp, 13

 RD117_MBED/MAX30102/MAX30102.cpp, 14
 RD117_MBED/MAX30102/MAX30102.h, 17
 RD117_MBED/algorithm/algorithm.cpp, 5
 RD117_MBED/algorithm/algorithm.h, 8
 RD117_MBED/main.cpp, 12
 REG_FIFO_CONFIG
 MAX30102.h, 18
 REG_FIFO_DATA
 MAX30102.h, 18
 REG_FIFO_RD_PTR
 MAX30102.h, 18
 REG_FIFO_WR_PTR
 MAX30102.h, 18
 REG_INTR_ENABLE_1
 MAX30102.h, 19
 REG_INTR_ENABLE_2
 MAX30102.h, 19
 REG_INTR_STATUS_1
 MAX30102.h, 19
 REG_INTR_STATUS_2
 MAX30102.h, 19
 REG_LED1_PA
 MAX30102.h, 19
 REG_LED2_PA
 MAX30102.h, 19
 REG_MODE_CONFIG
 MAX30102.h, 19
 REG_OVF_COUNTER
 MAX30102.h, 19
 REG_PART_ID
 MAX30102.h, 19
 REG_PILOT_PA
 MAX30102.h, 20
 REG_REV_ID
 MAX30102.h, 20
 REG_SPO2_CONFIG
 MAX30102.h, 20
 REG_TEMP_CONFIG
 MAX30102.h, 20
 REG_TEMP_FRAC
 MAX30102.h, 20
 REG_TEMP_INTR
 MAX30102.h, 20

 true
 algorithm.h, 10

uch_dummy
 main.cpp, [14](#)
uch_spo2_table
 algorithm.h, [12](#)